

ST6-Realizer w praktyce

Czyli: zostań "malarzem" programów - kontroler napięcia akumulatora samochodowego, część 1

O programie ST6-Realizer pisaliśmy już na łamach EP kilkakrotnie. Jak wynika z listów nadsyłanych do redakcji wielu Czytelników nie potrafi poradzić sobie z "oswojeniem" programu, co powoduje, że szybko się do niego zniechęcają. A szkoda! ST6-Realizer jest doskonałym narzędziem do szybkiej realizacji prostszych projektów na mikrokontrolerach ST62, pozwala także wkroczyć w ich „tajemniczy“ świat osobom nie znającym żadnego języka programowania! Aby ułatwić zrozumienie sposobu projektowania przy pomocy ST6-Realizera przedstawiamy bardzo prostą aplikację, wykonaną przy jego pomocy - jest to kontroler napięcia akumulatora samochodowego. W pierwszej części artykułu przedstawimy pierwsze trzy etapy projektowania. Za miesiąc przeanalizować pracę procesora i zaprogramować go.

Realizację projektu podzielono na pięć zasadniczych etapów:

- ✓ opracowanie koncepcji urządzenia,
- ✓ zaprojektowanie konstrukcji elektronicznej,
- ✓ „narysowanie“ programu,
- ✓ symulacja,
- ✓ zaprogramowanie mikrokontrolera

Ponieważ analiza poszczególnych etapów tworzenia projektu jest stosunkowo łatwa do przeprowadzenia, omówimy je kolejno.

Etap 1 - koncepcja

Urządzenie ma za zadanie monitorować w sposób ciągły napięcie w instalacji samochodowej i sygnalizować jego trzy stany (zgodnie z rys. 1):

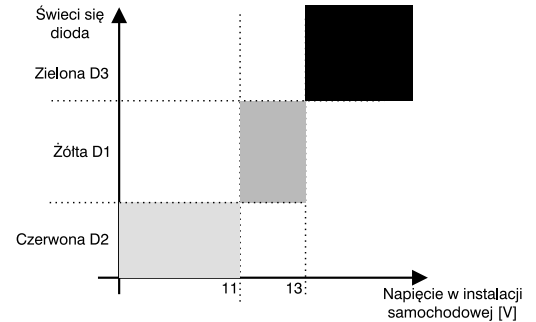
- Napięcie wysokie, powyżej 13V (przyjęto, że oznacza to bardzo dobrą kondycję akumulatora). Stan ten jest sygnalizowany świeceniem zielonej diody LED.
- Napięcie średnie, w przedziale pomiędzy 11 i 13V (przyjęto, że oznacza to poprawną pracę akumulatora).
- Napięcie niskie, poniżej 11V (przyjęto, że kondycja akumulatora jest zła).

Zasilanie monitora zasilania powinno pochodzić także z instalacji elektrycznej samochodu.

Przyjęte wartości napięć nie muszą być optymalne dla akumulatorów samochodowych, służą tutaj tylko jako ilustracja omawianego przykładu.

Etap 2 - konstrukcja elektryczna

Ponieważ do wykonania projektu potrzebne będą tylko 4 wyprowadzenia I/O mikrokontrolera (dla 3 diod LED i jedno wejście



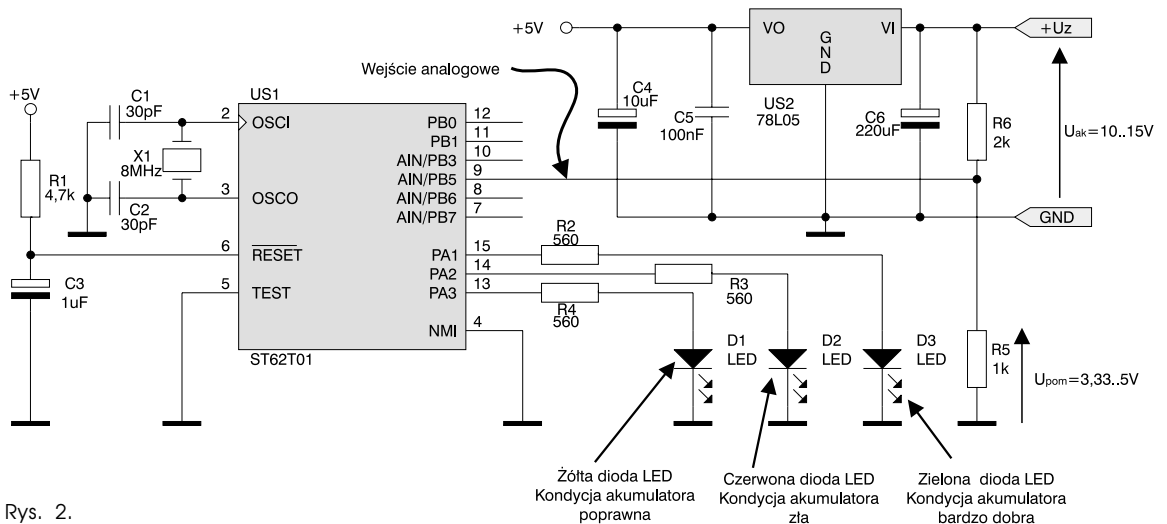
Rys. 1.

analogowe), możliwe było zastosowanie jednego z najmniejszych procesorów rodziny ST62 - ST62T01. Jest on montowany w obudowie DIL16, która pozwala wykorzystać 9 dwukierunkowych, uniwersalnych pinów I/O, z czego 4 mają możliwość pracy jako wejścia analogowe.

W wyniku analizy założeń z Etapu 1 powstał prosty schemat elektryczny, który przedstawiamy na rys. 2.

Elementy C1, C2 i X1 spełniają rolę generatora wzorcowego, który taktuje rdzeń mikrokontrolera. Elementy R1, C3 zapewniają zerowanie procesora po włączeniu zasilania. Układ US2 wraz z towarzyszącymi elementami spełnia rolę stabilizowanego zasilacza, o napięciu wyjściowym 5V. Diody D1..3 sygnalizują wynik pomiaru.

Rezystory R5, R6 odpowiadają za podział napięcia wejściowego, zapobiegając uszkodzeniu obwodów wejściowych układu US1. Zastosowanie tego dzielnika było niezbędne, ponieważ maksymalne dopuszczalne napięcie na wejściu procesora jest praktycznie równe napięciu zasilania. Zastosowanie dziel-

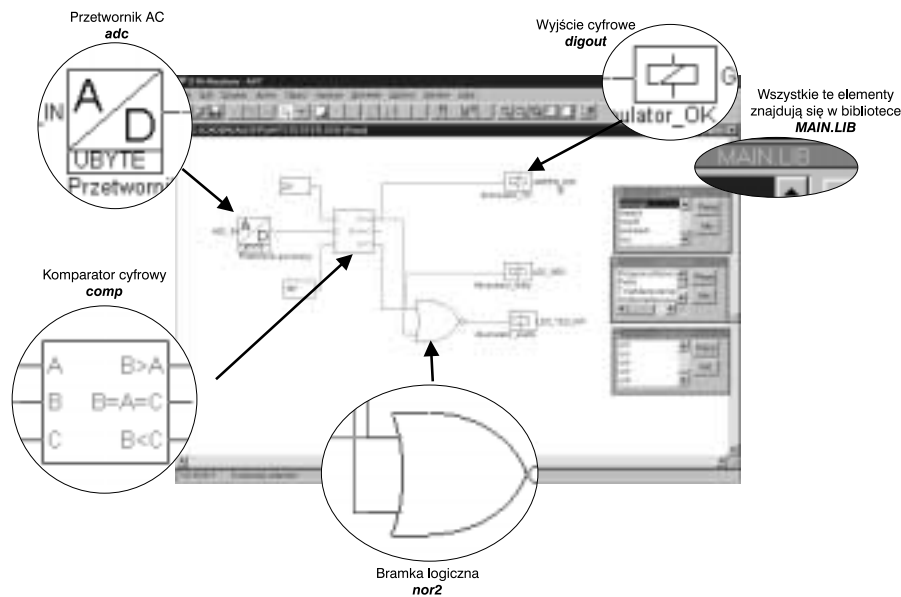


Rys. 2.

Żółta dioda LED
Kondycja akumulatora poprawna

Czerwona dioda LED
Kondycja akumulatora zła

Zielona dioda LED
Kondycja akumulatora bardzo dobra



Rys. 3.

nika rezystorowego o współczynniku podziału 1:3 gwarantuje, że jeżeli napięcie w instalacji samochodu nie przekroczy 15V procesor nie ulegnie uszkodzeniu.

Przyjęte przez autora przyporządkowania wyprowadzeń do funkcji jest zupełnie przypadkowe. Zarówno same procesory, jak i ST6-Realizer umożliwiają dowolne przypisanie funkcji do wybranych wyprowadzeń, z jednym tylko zastrzeżeniem - jeżeli wybrane wejście ma być skonfigurowane jako analogowe konstrukcja docelowego procesora musi to umożliwiać.

W prezentowanym przykładzie oznacza to, że rolę wejścia pomiarowego może spełniać dowolne wyprowadzenie spośród PB3, PB5...7.

Etap 3 - rysujemy program

Ponieważ funkcja spełniana przez projektowane urządzenie jest bardzo prosta, do narysowania programu wystarczy kilka prostych elementów bibliotecznych, tzn. przetwornik A/C, komparator cyfrowy, dwie stałe typu UBYTE, jedna bramka logiczna NOR i wyjścia cyfrowe (odpowiedniki pinów). Wszystkie te elementy wchodzi w skład standardowej biblioteki ST6-Realizera. Pokazano je na rys. 3.

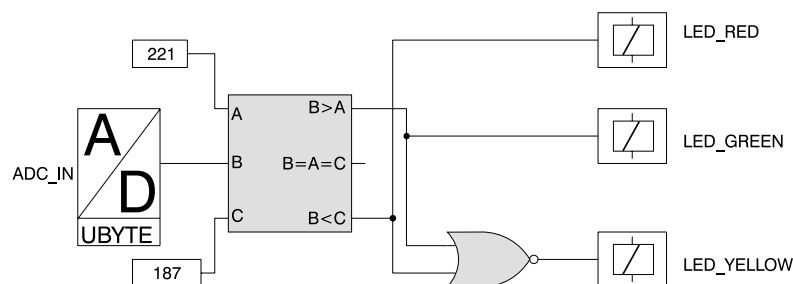
Na rys. 4 przedstawiono schemat stanowiący podstawę programu. Tworzenie projektu rozpoczynamy od stworzenia pliku „przewodnika“ (w menu: *File/New Project*). Następnie tworzymy plik ze schematem (w

menu: *File/New*) i możemy przystąpić do rysowania schematu.

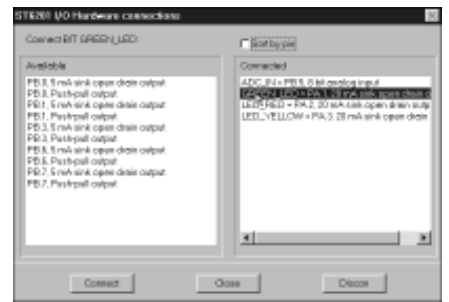
Po rozmieszczeniu na planszy elementów pobieranych z bibliotek (w menu: *Object/Library Symbol*) i wykonaniu pomiędzy nimi połączeń należy zdecydować dla którego układu z rodziny ST62 projekt tworzymy. W menu *Options/Select Hardware* wskazujemy plik o rozszerzeniu DLL zawierający opis wybranego układu. Następnie przypisujemy wyprowadzenia wykorzystywane w projekcie do fizycznych wyprowadzeń wybranego mikrokontrolera. Najłatwiej jest to zrobić poprzez dwukrotne kliknięcie na wybranym wyprowadzeniu, co spowoduje wyświetlenie się okna jak na rys. 5. Z jego lewej strony wyświetlana jest lista wolnych wyprowadzeń, z prawej strony - lista pinów już „podłączonych“.

Ostatnim krokiem jest sprawdzenie, czy projekt „zmieści“ się w wybranym procesorze, co robimy poprzez wybranie *Go* opcji *Analyse*. Jeżeli wszystko przebiegło pomyślnie na ekranie monitora zobaczymy bardzo sympatyczny komunikat (rys. 6).

Na wejście ADC_IN przetwornika A/C podawane jest napięcie wejściowe (z dzielnika R5, R6 - rys. 2). Po przetworzeniu na binarne słowo 8-bitowe (UBYTE) wynik podawany jest na wejście B komparatora cyfrowego. Komparator ten ma dwa dodatkowe wejścia A i C, które spełniają w prezentowanym urządzeniu rolę punktów odniesienia. Stała na wejściu A określa górny próg



Rys. 4.



Rys. 5.

referencyjny układu pomiarowego, wyznaczając poziom 13V. Podanie na wejście A stałej o wartości 221 wynika z następującego wyliczenia:

- wartość napięcia wejściowego U_{we} wynosi dla tego progu 13V;
- jeżeli napięcie mierzone ma wartość 13V, to na wejściu PB5 procesora pojawia się napięcie $13V:3$ (wynika ze stopnia podziału dzielnika wejściowego $R5, R6$) = 4,33V,
- ponieważ rozdzielczość przetwornika wynosi 8 bitów, to napięciu wejściowemu 4,33V odpowiada na wyjściu przetwornika liczba binarna: $(4,33 * 255) / 5 = 221$.

Wychodząc z założenia, że wejście C komparatora wyznacza dolny próg odniesienia (dla 11V) możemy wykonać podobne wyliczenie: $(3,66 * 255) / 5 = 187$. Stąd wynika wartość słowa przypisanego na stałe do wejścia C.

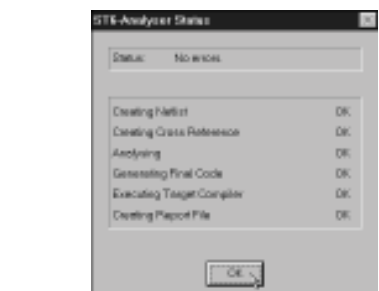
Ponieważ wyjścia zastosowanego komparatora wskazują tylko dwa interesujące nas stany (napięcie poniżej dolnego lub powyżej górnego progu) konieczne było zastosowanie dodatkowo bramki NOR, która steruje świeceniem żółtej diody LED. Dioda ta zapala się tylko wtedy, gdy napięcie wejściowe nie przekroczyło żadnego z zadanych progów.

Piotr Zbysiński, AVT

Program ST6-Realizer oraz komplet narzędzi i danych katalogowych dla mikrokontrolerów ST62, znajdują się na płycie CD-EP4.

Uwaga! Prosimy o nadsyłanie pomysłów, które chcielibyście zrealizować przy pomocy ST6-Realizera. Najciekawsze z nich wykonamy dla Was, prezentując na łamach EP w jaki sposób zostało to wykonane.

Komplet plików tworzących projekt TES-TER znajduje się w Internecie pod adresem: www.avt.com.pl/avt/ep/ftp.



Rys. 6.